

# Simulating an underwater vehicle self-correcting guidance system with Simulink

FAN Hui\* (范 辉), ZHANG Yu-wen (张宇文), LI Wen-zhe (李文哲)

College of Marine Engineering, Northwestern Polytechnical University, Xi'an 710072, China

**Abstract:** Underwater vehicles have already adopted self-correcting directional guidance algorithms based on multi-beam self-guidance systems, not waiting for research to determine the most effective algorithms. The main challenges facing research on these guidance systems have been effective modeling of the guidance algorithm and a means to analyze the simulation results. A simulation structure based on Simulink that dealt with both issues was proposed. Initially, a mathematical model of relative motion between the vehicle and the target was developed, which was then encapsulated as a subsystem. Next, steps for constructing a model of the self-correcting guidance algorithm based on the Stateflow module were examined in detail. Finally, a 3-D model of the vehicle and target was created in VRML, and by processing mathematical results, the model was shown moving in a visual environment. This process gives more intuitive results for analyzing the simulation. The results showed that the simulation structure performs well. The simulation program heavily used modularization and encapsulation, so has broad applicability to simulations of other dynamic systems.

**Keywords:** Simulink; self-correcting ahead angle arithmetic; vehicle acoustic homing system; modularization

**CLC number:** TP273

**Document code:** A

**Article ID:** 1671-9433(2008)03-0205-07

## 1 Introduction

Underwater vehicle guidance research by means of simulation has advantages of validating the rationality of guidance design just by the computer-aided calculation and adjusting guide method and parameters based on simulation results, thereby saving test expenses available.

Many guide methods can be applied to the guidance design, such as nil lead angle guide, fixed lead angle guide, variable lead angle guide and parallel approach guide, etc. These guide algorithms above have common characteristics that they all obtain continuous dynamic guide rules by certain mathematic relations based on angular or distance information of vehicles and targets. The research on these guide algorithms just need to implement the mathematics description of motion models and algorithms by simple program<sup>[1]</sup>. However, applying the self-correcting lead angle guide algorithm is much more different. This algorithm is a kind of discrete logic based on event-driven, once coupling with motion models of

vehicle and target, and it will present characteristics of hybrid systems. The following challenges of modeling and simulation come from two aspects: one is the description of discrete logical system. It is not a standardized mathematic system, involving many conditional embranchments and transfers, and involved with saving and rereading of several history states; the other one involves the synchronous coordination and link between the two parts of continuous and discrete systems.

Using general programming language, such as C, C++, Basic, etc. in simulation research, the realization of complicated mathematic algorithm is certainly the first obstacle, and secondly, the graphic display of simulation results also need a great deal of programming work, and the accuracy of that is even not assured easily. The Matlab software produced by MathWorks company has powerful mathematics calculation functions<sup>[4]</sup>. It has excellent numerical calculation ability and data visualization ability, so it's broadly used in automatic control, digital signal process, dynamic system simulation etc. To exactly import the control system's complex model to the computer for analysis and simulation, MathWorks

Received date: 2008-03-05.

\*Corresponding author Email: aooooaooooo@126.com

offered a new tool of graphics input and simulation for the control system modeling: Simulink. It provides an integrated work environment to realize dynamic system model construction and simulation. As an important part of Matlab, Simulink enlarges Matlab's functions. Simulink has relatively independent functions and methods, and supplies a friendly graphics user interface (GUI). Its model is denoted by frame figures composed of models, and can realize visualization construction. Simulink is a powerful tool for dynamic system model simulation<sup>[5]</sup>.

This paper takes advantages of Simulink at modeling, simulation and graphical display, aiming at guidance simulation, in which self-correcting lead angle algorithm is applied in underwater vehicles, and a type of incorporate simulation structure, including both mathematical results and synchronous visual scene output, is explored. In this paper, the models of vehicle kinematics, target kinematics and vehicle-target relative motion will be established to

complete the encapsulation of sub-modules in Simulink environment. The StateFlow module will be used in Simulink to realize the guidance algorithm based on the state flow and trigger mechanism. Meanwhile, VRML will be used to create 3-D model of vehicle and target, and implement visual scene synchro-display along with mathematical output.

## 2 Simulation model of guidance

The flow chart of integrated guidance simulation progress is shown in Fig.1. The target azimuth angle information is acquired according to the vehicle-target relative motion, the expected attitude angle is calculated via self-correcting lead angle guide algorithm, which is then applied to the close-loop control system and compared with currently actual attitude to decide rudder operation. The model of vehicle motion responds to rudder angle change to form new attitude, the circle is repeated until vehicle is led to target.

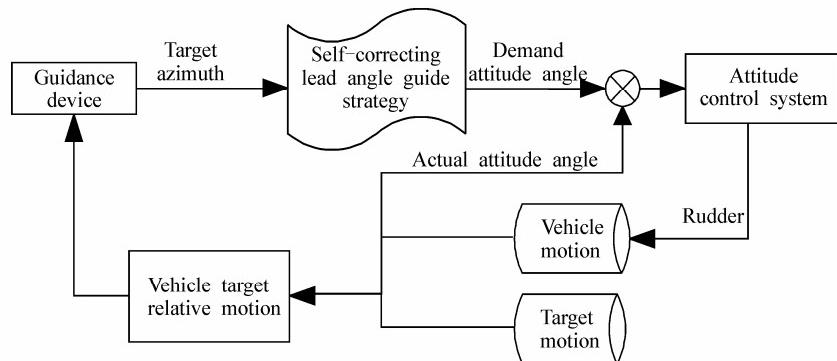


Fig. 1 Principle layout of integrated guide system

### 2.1 Mathematical models of motion

While underwater vehicles move in horizontal plane without roll, the dynamics equations are:

$$\begin{cases} \frac{d\omega_y}{dt} = a_1\omega_y + a_2\beta + a_3\delta_r, \\ \frac{d\beta}{dt} = a_4\omega_y + a_5\beta + a_6\delta_r, \end{cases} \quad (1)$$

where  $\omega_y$ ,  $\beta$ , and  $\delta_r$  are yaw angle rate, sideslip angle and vertical rudder angle, respectively.  $a_i$  ( $i=1,\dots,6$ ) are coefficients depending on the integrated figure of the vehicle.

When the vehicle moves in horizontal plane without roll, the transverse kinematics equations are:

$$\begin{cases} \dot{\psi}_t = \omega_y, \\ \dot{\Psi}_t = \dot{\psi}_t - \dot{\beta} = (1-a_4)\omega_y - a_5\beta - a_6\delta_r, \\ \dot{X}_t = V_t \cos \Psi_t, \\ \dot{Z}_t = V_t \sin \Psi_t, \end{cases} \quad (2)$$

where  $\psi_t$ ,  $\Psi_t$ ,  $V_t$ ,  $X_t$ , and  $Z_t$  are yaw angle, trajectory departure angle, vehicle velocity, and position of vehicle in inertia coordinate frame, respectively.

The kinematics equations of target in horizontal plane are:

$$\begin{cases} \dot{\psi}_m = \omega_m, \\ \dot{X}_m = V_m \cos \psi_m, \\ \dot{Z}_m = V_m \sin \psi_m, \end{cases} \quad (3)$$

where  $\psi_m$ ,  $\omega_m$ ,  $V_m$ ,  $X_m$ , and  $Z_m$  are trajectory departure

angle and yaw angle rate of target, target speed and the position of target in inertia coordinate frame, respectively.

The vehicle-target relative motion relationship is:

$$\begin{cases} \dot{r} = V_m \cos(q - \psi_m) - V_t \cos(q - \psi_t), \\ \dot{q} = \frac{1}{r} [V_t \sin(q - \psi_t) - V_m \sin(q - \psi_m)], \end{cases} \quad (4)$$

where  $r$  is the relative distance between vehicle and target,  $r = \sqrt{(X_m - X_t)^2 + (Z_m - Z_t)^2}$ ;  $q$  is relative azimuth angle, which is defined as  $q = \arctan(-(Z_m - Z_t)/(X_m - X_t))$ .

The simultaneous equations (1)~(4) represent the definition of mathematical model for vehicle guidance in horizontal plane. It is a set of equations that is composed of 11 first order differential equations. Given the initial condition associated with the vehicle and target and joined with guide strategy, all the flight date during guidance process can be solved.

The simulation subsystem that simulates the characteristics of vehicle and target is built in Simulink environment, as shown in Fig.1, which has implemented the encapsulation of the vehicle motion model, target motion model and the model of relative motion between them.

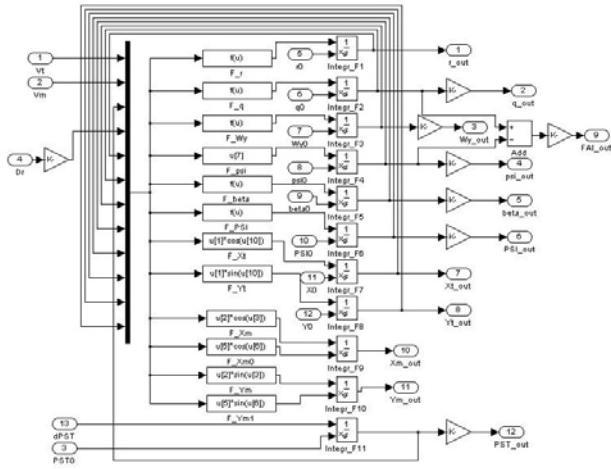


Fig.1 Model of vehicle-target motion subsystem

## 2.2 Realization of self-correcting lead angle guidance algorithm

As shown in Fig.2, we assume that the guidance beam sector is composed of 11 separate beam segments, numbered by -5 to 5, every segment overlays a sector

with certain degree.

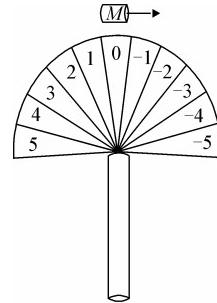


Fig.2 Sketch map of guidance beam sector setup

When target gets into the guidance beam area, No.0 beam will be used as benchmark axis aiming at it at first. After this, the benchmark axis location in beam area will change continuously according to the relative motion situation between vehicle and target, thus the lead angle is adjusted automatically to be proper and opportune, then the demand attitude of vehicle is applied to the control system. The detailed process is shown in Fig.3.

The keystone of this algorithm is that firstly, record the sequence number of the segments in which the target and benchmark axis appeared. Adjust the benchmark location continuously by estimating the change trend between the numbers, so as to establish the demanded lead angle, which will lead vehicle course head to the encounter point with target.

Let  $B_N(k)$  and  $Z_N(k)$  respectively be the sequence number of the segments in which the target and benchmark axis appeared, get  $B_N(k)$  by estimating which segment's angle range does the target azimuth (obtained by sampling continuous system) belong to in every sample period. Let  $Z_N(0)=0$ , and define  $D_{st}(k)$  as the warp state between target and benchmark axis:

$$D_{st}(k) = B_N(k) - Z_N(k). \quad (5)$$

Then, the adjustment law of benchmark axis can be described as:

$$Z_N(k+1) = \begin{cases} Z_N(k) + 1, & D_{st}(k+1) > D_{st}(k); \\ Z_N(k), & D_{st}(k+1) = D_{st}(k); \\ Z_N(k) - 1, & D_{st}(k+1) < D_{st}(k). \end{cases} \quad (6)$$

Let the lead angle of vehicle be  $\delta(k)$ , and every beam segment overlay angular degree by  $\vartheta$ , then the adjustment strategy of lead angle can be described as:

$$\delta(k+1) = \begin{cases} (B_N(k+1) - Z_N(k+1)) * \vartheta, & D_{st}(k+1) \neq D_{st}(k); \\ D_{st}(k) * \vartheta, & D_{st}(k+1) = D_{st}(k). \end{cases} \quad (7)$$

Let  $\psi(k)$  be the yaw angle of vehicle, then

$$\psi(k+1) = \psi(k) + \delta(k+1). \quad (8)$$

Variables to be defined in StateFlow module is listed in Table 1. The program will first obtain single beam segment width ( $B_{SA}$ ), target azimuth angle ( $A_{zA}$ ) and current attitude of vehicle ( $P_{AA}$ ) from extra Simulink models, then estimate the current sequence number of target location ( $B_N$ ). The partial program for this step is shown in Fig.4.

Then, the program will estimate the relative location state between benchmark axis and target according to the fore-step  $Z_N$ , and then, judge the opposite moving trend between target and benchmark axis according to the fore-step  $D_{st}$ , and thereby decide the moving strategy of benchmark axis. In this process, an edge restriction has to be added at the benchmark axis adjustment. When  $Z_N$  is out of  $[-4, 4]$ , the adjustment will not be allowed. After the above process, the program will record the current warp distance  $D_{st}(k+1)$  between benchmark axis and target, which is convenient for comparing in the next circle. This part is shown in Fig.5.

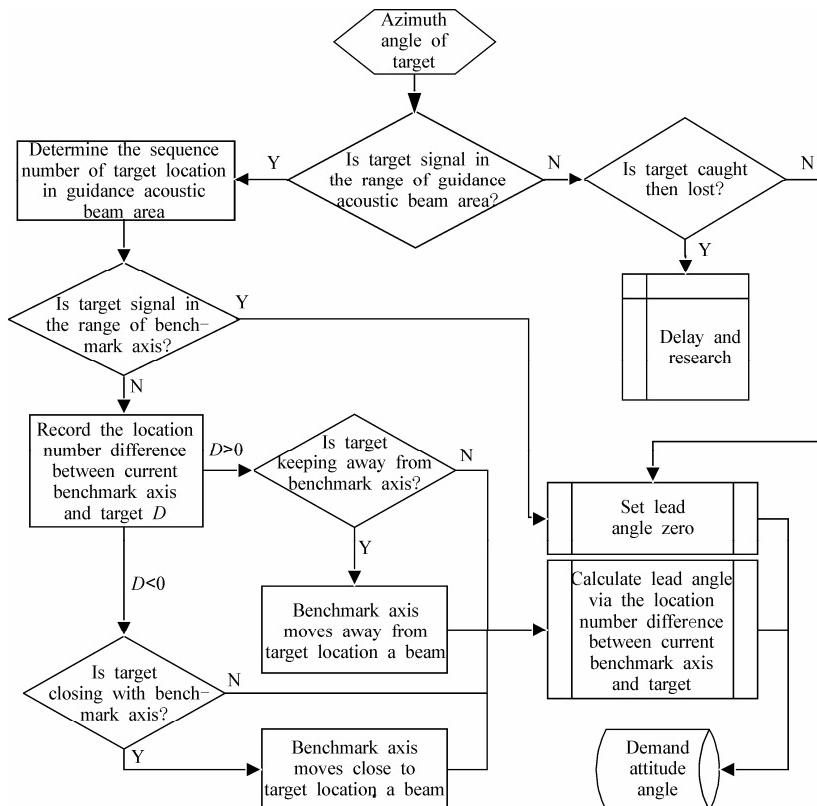


Fig.3 Principle layout of self-correcting lead angle guidance algorithm

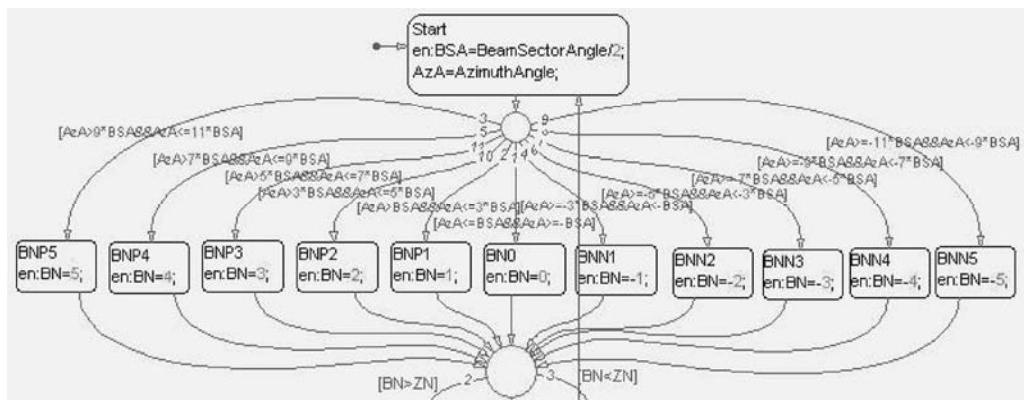


Fig.4 Partial StateFlow program ( $B_N$  estimation)

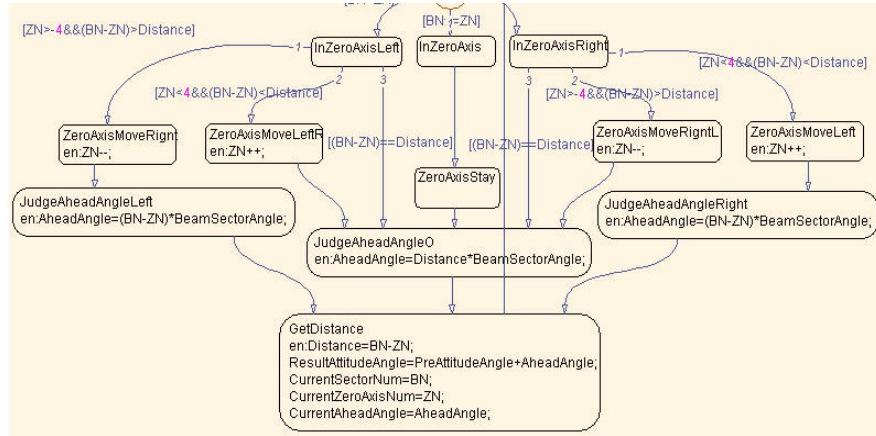


Fig.5 Partial StateFlow program (decision-making at benchmark axis adjustment and lead angle)

**Table 1 Variables defined in StateFlow program**

Variables	Type
$B_{SA}$	Input
$A_{zA}$	Input
$P_{AA}$	Input
$M_W$	Event
$R_{AA}$	Output
$C_{SN}$	Output
$C_{ZA}$	Output
$C_{AA}$	Output
$B_N$	Local
$Z_N$	Local
$D_{st}$	Local
$A_{hA}$	Local

The triggering of StateFlow program is implemented by an extra trigger signal ( $M_W$ ). The program outputs current target location number ( $C_{SN}$ ), current benchmark axis location number ( $C_{ZA}$ ), current lead angle decision value ( $C_{AA}$ ) and current attitude angle decision of vehicle ( $R_{AA}$ ) in every work circle, which are offered for the following simulation analyses.

### 2.3 Synchronous visual scene display

Simplex mathematical simulation can offer accurate data curve, but it is not intuitionistic enough. When researchers analyze the simulation results, they have to ascertain the whole simulation progress by an integrated thought of the object's location and circumgyration. The synchronous visual scene display is designed along with the mathematical simulation to embody the simulation progress, which can offer more abundant and intuitionistic vision information, and thereby increase the design efficiency of vehicle controller and guidance device<sup>[6][8]</sup>.

Making use of the VR Sink module in Simulink, the visual scene display can be implemented for mathematical simulation results conveniently. The VR module supports the 3-D model of vehicle and target

created by VRML, and it receives the data relative to movement provided by extra model in Simulink, and displays the visual scene synchronously.

To be convenient for observing the operation of guidance device, independent model of guide beam area was created and put upon the nose of vehicle. For the reason that the overlay degree of single beam segment must be adjusted, the 3-D models need to be driven include 11 radials indicating guide beam besides vehicle and target. Therefore, this sub model looks a little more complicated, see Fig. 6.

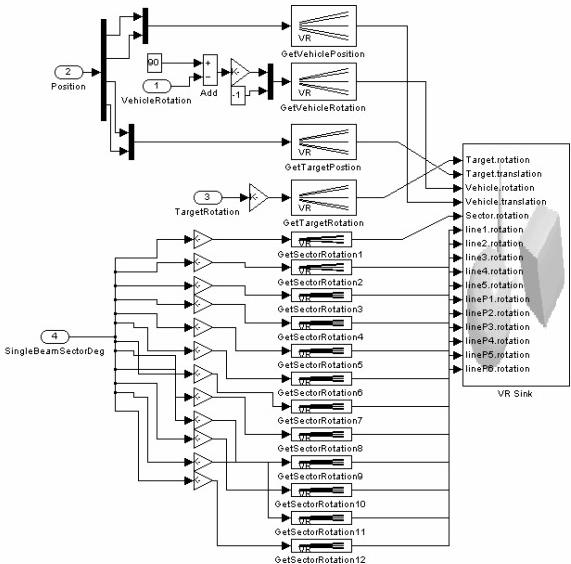


Fig. 6 Synchronous visual scene display sub-model

Summarize above discussion, the integrated Simulink model is composed of several sub modules, including initial module, Vehicle-Target subsystem, self-correcting lead angle guide algorithm module, data output and display module and synchronous visual scene display module. The entire simulation

system is shown in Fig. 7.

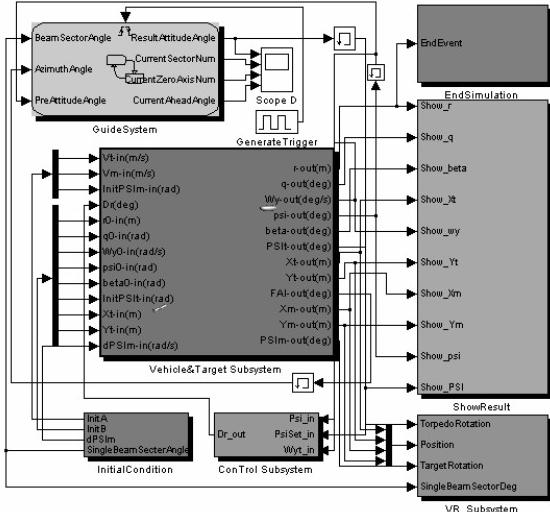


Fig. 7 Integrated simulation model in Simulink

### 3 Simulation results

Assuming vehicle makes a uniform motion, the target makes uniform straight motion at initial phase, and then makes a convolute maneuver.

Suppose vehicle moves at speed  $V_t=21$  m/s, the initial condition associated with vehicle is

$$\beta(0)=\omega_y(0)=\delta_r(0)=X_t(0)=Z_t(0)=0, \psi_t(0)=0.$$

The target velocity is  $V_m=8$  m/s, the initial condition about target is

$$X_m(0)=Z_m(0)=707.1068 \text{ m}.$$

Namely, initial distance is 1 000 m, initial relative azimuth angle is 45°. When simulation time  $t<20$  s, target moves straight at uniform speed without yaw; when  $t>20$  s, target does a convolute maneuver at angular rate 0.2 rad/s.

According to the work frequency of vehicle's self-guide system, a signal was generated to trigger the guide algorithm sub module to work. The setup of this signal in simulation is that it is a pulse with a period of 0.05 s; its width is 50% of period (namely the sample time is 25 ms). The solver is set to ode45 (Dormand-Prince) variable-step, the maximum step size is 0.15 s, and the minimum step size is 0.1 s. The simulation results are shown in Fig.8 and Fig.9.

From the simulation results shown in Fig.8, it can be seen that an approximately straight motion path could be obtained even at the situation of target maneuver;

Fig.9 shows that the diagrammatic program based on state flow and event-driven mechanism can realize the self-correcting lead angle guide algorithm successfully.

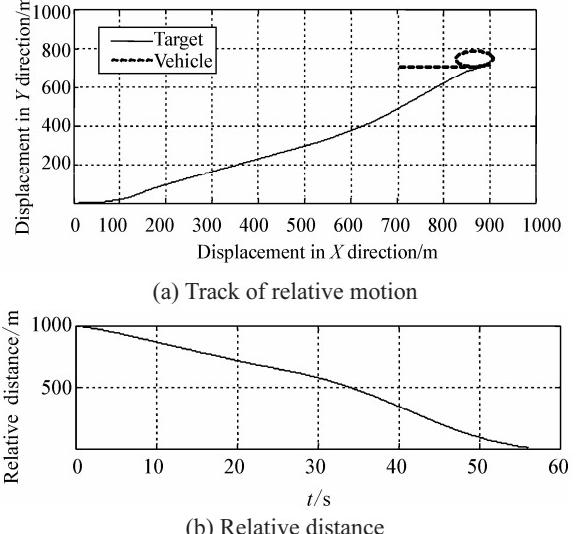


Fig.8 Relative motion between vehicle and target

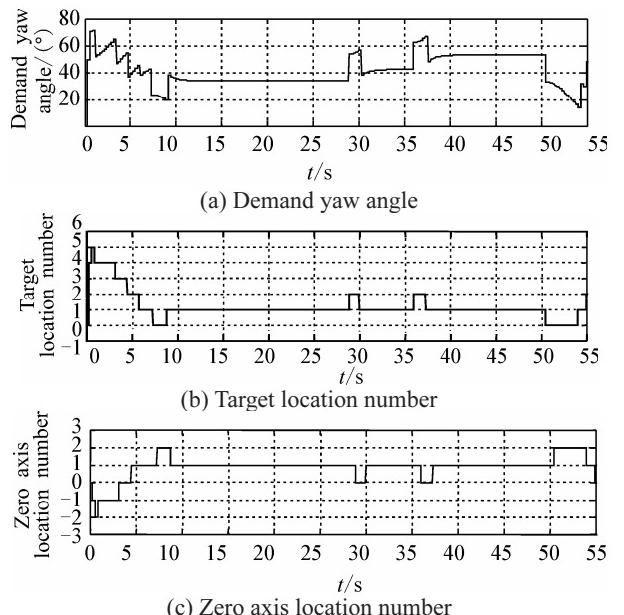


Fig.9 Demanded yaw angle, real-time sequence number of target location and benchmark axis

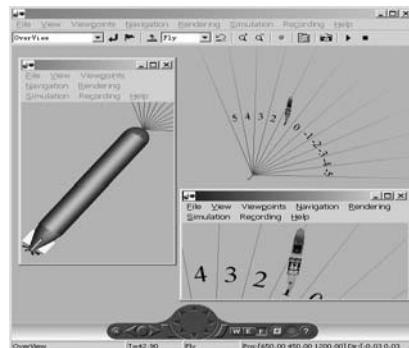


Fig.10 Synchronous visual scene display ( $t=43.0$  s)

Fig.10 shows the synchronous visual scene display at simulation time  $t=43.0$  s. This display is composed of three view windows for overview, vehicle view and target view respectively.

## 4 Conclusions

The underwater vehicle guidance incorporate simulation is completed with both mathematical output and visual scene display in this paper, in which the vehicle adopts self-correcting lead angle guide algorithm. Conclusions can be drawn as follows:

- 1) The self-correcting lead angle guide algorithm has special recursive and discrete characteristic. Modeling it with text describing program language will have to be involved with a mass of condition branches and circles, which will bring difficulties at program writing and debugging. In this paper, StateFlow module based on state flow mechanism is used to model this algorithm. It is totally graphic program, simple in operation, clear in layers and effective in program. The state flow could be observed in debugging process, and it is convenient for program maintenance.
- 2) The trigger execution method can be adopted when integrating a discrete subsystem into a continuous system. It works like this: the discrete system samples signals coming from the continuous system and then executes inner algorithm and outputs results. In this way, the two different types of systems couple to work and don't affect each other in solving process.
- 3) Utilizing the VR Sink module in Simulink to implement synchronous visual scene display along with mathematical simulation, extra program work is not needed too much, only the mathematical results are needed to drive the 3-D model created by VRML moving in visual scene. The synchronous visual scene display can be a more intuitionistic inspective tool in simulation research.
- 4) Taking advantage of Simulink, relative motion of vehicle and target, guide algorithm, initial module, termination check module and results display module are encapsulated into separate subsystems and joined together, each subsystem can be employed independently and displaceablely, that raises the

adaptability and has a general reference bearing to the simulation of other underwater vehicles.

## References

- [1] YE Lingwei, LI Shuhua, WANG Shouquan. Study and simulation of torpedo's guidance and trajectory[J]. Science & Technology Information, 2007, 18: 50-51(in Chinese).
- [2] TONG Yong, QU Hongxing, CAI Jiefeng. Analysis on the before-revise guidance regulation trajectory simulation[J]. Command Control & Simulation, 2006(3): 104-105(in Chinese).
- [3] LIU Enxiang; HUANG Changqiang; ZOU Qiang. Research on guidance method and ballistic simulation of underwater missiles[J]. Fire Control & Command Control, 2004(4): 57-59.
- [4] YANG Chunquan; NI Shihong; WU Wenhai; SHEN Chunlin. Matlab-based simulation & study on the pursuit-interception control of A/C[J]. Computer Simulation, 2003(2): 17-19(in Chinese).
- [5] CHAI Lin, FANG Qun. The Simulation of torpedo's guidance trajectory based on MATLAB/Simulink[J]. Acta Simulata Systematica Sinica, 2003(2): 79-82.
- [6] PAN Jinchuan, JIANG Changsheng. A new intelligent navigation law and its 3-D visualization simulation[J]. Mechanical Engineering & Automation, 2005(5): 10-12(in Chinese).
- [7] WANG Xinxiao, HUANG Jianguo, ZHANG Qunfei, LV Yuan. The design of the integrated underwater vehicle homing system simulation software[J]. Computer Engineering and Applications, 2003(3): 213-215(in Chinese).
- [8] HUA Feng, HUANG Jianguo, LI Xiaodong. Implementation of 3-D animation in autonomous underwater vehicles homing system[J]. Computer Simulation, 2000(2): 4-7 (in Chinese).



**FAN Hui** was born in September 1981. He is a PhD candidate at Northwestern Polytechnical University, majoring in weapon system collectivity design theory and method.



**ZHANG Yu-wen** was born in July 1946. He is a professor at Northwestern Polytechnical University, and a doctoral supervisor, majoring in weapon system and expert engineering, firing theory and technology, trajectory and control, etc.



**LI Wen-zhe** was born in March 1974. He is an instructor, and a PhD candidate at Northwestern Polytechnical University, majoring in weapon system trajectory and combat efficiency research and teaching work.